AMENDMENTS TO THE SPECIFICATION:

Page 1, amend paragraph [0001] as:

[0001] This invention relates generally to hot-swap devices applicable to the known integrated drive electronics (IDE) bus and, more particularly, it relates to a hot-swap device applicable to the conventional IDE bus without changing the standard driver.

Page 1, amend paragraph [0002] as:

[0002] A conventional IDE bus usually doesn't support the hot-swap function and, meanwhile, the standard driver cannot sense any change of a storage hardware via the IDE bus. Therefore, for achieving ~~performance of~~ the hot-swap function through the conventional IDE bus ~~in this case~~, an extra nonstandard driver is needed for scanning the IDE bus back and forth trying to find out if any new storage device is joined. However, under such a situation, the standard driver must be substituted by the nonstandard one that would probably deteriorate the compatibility in the computer system

Pages 1-2, amend paragraph [0003] as:

[0003] When the change of the storage hardware cannot be detected through the IDE bus by a computer operating system (OS), troubles may come up as the following:

(a) Should the OS have failed in detecting a storage device on the IDE bus when booting, then it wouldn't be able to find out any newly joined storage hardware such as a hard disk drive thereinafter~~, a hard disk drive for example~~.

(b) When the OS instructs to read or write through the IDE bus in the case of lacking any storage hardware, the situation becomes unexpectable to result in, for example, a "system down" in the worst case.

In view of the ~~abovesaid~~ above problems, this invention is to provide a feasible way that performs the hot-swap function via a conventional IDE bus without changing the standard driver

Page 2, amend paragraph [0006] as:

[0006] In order to realize the above objects ~~abovesaid object~~, the hot-swap device of this invention applicable to the ATA (AT Attachment) interface should comprise at least an IDE hard disk drive (HDD) controller for processing IDE instructions transferred from the ATA interface, wherein at least a program code is provided to the IDE hard disk drive controller to respond to the ATA interface with "a virtual storage device" in the case of lacking a real storage device connected to the IDE HDD controller via the ATA interface. In addition, [[or]] if the program code is executed to connect "a real storage device" to the IDE HDD controller via the ATA interface, connection of the IDE HDD controller with the ATA interface is disabled so that the real storage device will respond to the ATA interface directly.

Pages 2-3, amend paragraph [0007] as:

[0007] Moreover, the hot-swap device of this invention applicable to the ATA (AT Attachment) interface should comprise at least an IDE hard disk drive (HDD) controller for processing IDE instructions transferred from the ATA interface, wherein at least a

program code is provided to the IDE hard disk drive controller for the same to execute and respond to the ATA interface with "a virtual storage device" in the case of lacking a real storage device, or respond to the ATA interface with "a real storage device" if the program code is executed to connect the real storage device with the IDE HDD controller.

Page 3, amend paragraph [0012] as:

[0012] Fig. 4 shows an embodiment of a virtual ID table.

Page 3, amend paragraph [0012] as:

[0013] Fig. 5 shows an embodiment of a virtual partition table.

Pages 3-4, amend paragraph [0014] as:

[0014] As shown in Fig. 1—a first embodiment of this invention—the IDE (integrated drive electronics) hard disk drive (HDD) controller **10** of a hot-swap device is coupled with an ATA (AT Attachment) interface **20**, which is connected with a disk drive device **30** and a system bus **40** respectively. A memory **10a** of the IDE hard disk drive (HDD) controller **10** is provided at least with a program code for the IDE hard disk drive controller **10** to run and respond to the ATA interface **20** with "a virtual storage device" in the case of lacking a real storage device such as the disk drive device **30** connected to the IDE HDD controller. In addition, ~~for example the disk drive device 30, or~~ when the program code is executed to connect the "real storage device" with the IDE HDD controller **10** via the ATA interface, connection between the IDE HDD **10** and the ATA interface **20** is disabled while the "real storage device" will respond directly to the ATA interface instead.

Page 4, amend paragraph [0015] as:

[0015] Figs. 2A and 2B show a second embodiment of this invention. In Fig. 2A, the IDE hard disk drive controller **10** of the hot-swap device is coupled with the ATA (AT Attachment) interface **20** and the disk drive device **30** while the IDE hard disk drive controller **10** of the hot-swap device in Fig. 2B is coupled with the ATA interface **20** and a compact-flash storage device **32** respectively, wherein the ATA interface **20** is coupled with a PCMCIA/CF interface **50** connected with the system bus **40**. The memory **10a** of the IDE hard disk drive controller **10** is provided at least with a program code for the IDE hard disk drive controller **10** to run and respond to the ATA interface **20** with "a virtual storage device" in the case of lacking a real storage device, for example the disk drive device **30** or the compact-flash storage device **32** connected to the controller, or "a real storage device" otherwise.

Pages 4-5, amend paragraph [0016] as:

[0016] In a flowchart shown in Fig. 3, the IDE hard disk drive controller **10** is operated without connecting with any real storage device mentioned above. When booting a computer, the system bus **40** is supposed to deliver an ATA identify command (one of the IDE instructions) to the IDE hard disk drive controller **10** via the ATA interface **20** or via the PCMCIA/CF interface **50**. The ~~, then the~~ ATA interface **20**, and the IDE hard disk drive controller **10** would run the program code now to output ~~for output of~~ a virtual ID (identifier) table **60**. When the operating system is activated, it is supposed to send out an ATA identify command to the IDE hard disk drive controller **10**, which in

turn runs the program code to output ~~for output of~~ the virtual ID table **60**. Now, the operating system commands the IDE hard disk drive controller **10** to run the program code to read sectors of a partition table and give a virtual partition table **70**.

Page 5, amend paragraph [0017] as:

[0017] Fig. 4 shows an embodiment of a virtual ID table—a promulgated industrial standard already put into practice. A virtual ID table **60** shown in Fig. 4 is an embodiment cooperative with the hot-swap device of this invention under standard industrial regulations, wherein all the related numerals have been defined in hexadecimal codes for realizing a virtual storage device

Page 5, amend paragraph [0018] as:

[0018] Fig. 5 shows an embodiment of a virtual partition table—also a promulgated industrial standard already put into practice. A virtual partition table **70** shown in Fig. 5 is an embodiment cooperative with the hot-swap device of this invention under standard industrial regulations, wherein all the related numerals have been defined in hexadecimal codes for realizing a virtual storage device. Offset 4 in value "YY" of the virtual partition table **70** represents a partition mode ~~on account~~ of a practical application, wherein 00, 06, 0B, 0C, and 0F represent a partition mode of "Unknown", "FAT 16", "FAT 32", "FAT 32 28. 4G", and "Extend FAT 32" respectively.

Pages 5-6, amend paragraph [0019] as:

[0019] In order to fetch a correct address every time the operating system is to read or write, this invention takes the advantage of a specialty of the operating system, that is, the

operating system will rearrange the structure of the file system when a real HDD is newly added to a computer system equipped with no HDD-like storage device before. For working together in cooperation with the specialty of the operating system, when the HDD controller 10 runs to read the sectors of a virtual storage device, all the data in those sectors is responded with "0" except sector 0 that includes output of a virtual partition table 70, so that the operating system will admit the existence of a virtual storage device which is not yet formatted. And after the real storage device of a new HDD is added, the operating system will read the sectors of the new HDD once more for updating, and for assuring normal operation of an instruction, the operating system must have set the "attribute" of the HDD as "portable" in order to avoid delay of data-writing.